Gaussian Mixture Model

We assume that the dataset $D = \{x_n\}$ is generated by a probability distribution $P(x)$ that is just a weighted sum of K gaussians.

$$P(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x; \mu_k, \Sigma_k)$$

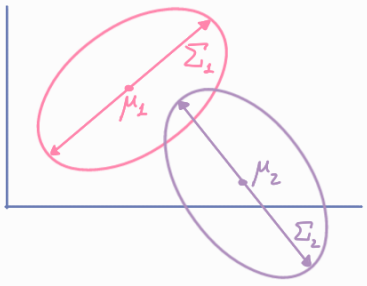
π_k: prior probability; μ_k: mean; Σ_k: covariance matrix

Total number of parameters of the model: $\underbrace{(\pi_k, \mu_k, \Sigma_k)}_{3} \times K$

Supposing K is known and that is an input for the model, we want to estimate the parameters.

Let's consider $K = 2$, $\pi_1 = 0.6$

$\pi_2 = 1 - \pi_1 = 0.4$



Given the parameters of the distribution we can easily generate a dataset: first we sample the gaussians and then we keep all the points.

π_1, μ_1, Σ_1
π_2, μ_2, Σ_2 $\} \longrightarrow D$

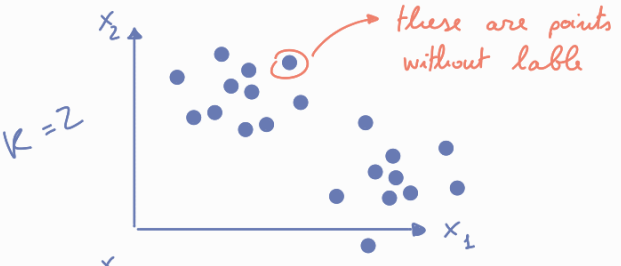
We are interested into the opposite

π_1, μ_1, Σ_1
π_2, μ_2, Σ_2 $\} \xleftarrow{?} D$

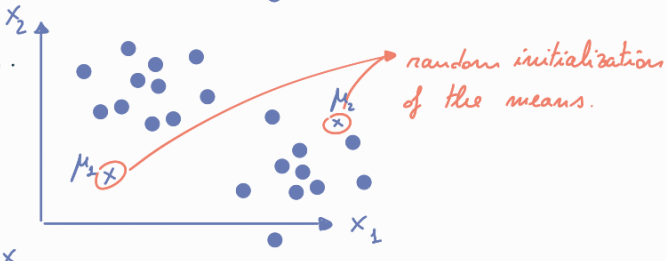
K-means

Simplify the problem: given a dataset $D = \{(x_n)\}$, given K, we want to compute the means for the k gaussians. This is called the k-means algorithm.
k-means is an iterative algorithm:

1. The input is a set of unlabeled data and a value for k

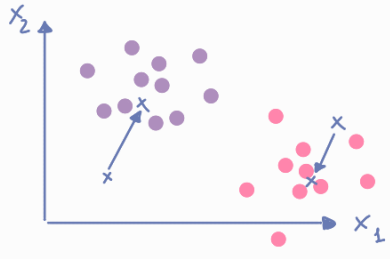


2. Start with a random initialization of the means.



3. Form the clusters: assign each point in the dataset to the closest mean.
For each point we compute the distance from every mean and assign the point to the right cluster.
After this step we have partitioned the dataset in K subsets.

4. Refine the means given the clusters.
Compute the mean of each cluster, that will become the new estimation for the previous mean



5. Repeat the process until we have no changes between two iterations (clusters and means remains the same).

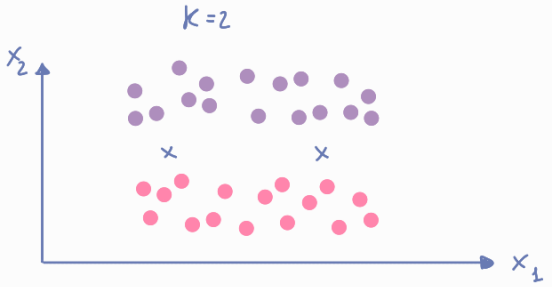
The convergence will always occur if the following two conditions are satisfied:

1. Every time you iterate and form a new cluster with a new mean, the sum of distances from each training sample to that training sample's group centroid is decreased;
2. There are only finitely many partitions of the training examples into K clusters.

Remarks on K-means

- The number of clusters K must be determined before hand;
- Sensitive to initial condition when few data is available;
- Not robust to outliers. Data very far from the real centroid may pull the mean far away;
- Does not account for the covariance matrix

Limitations mostly depends on the distance function



Even if you correctly guess K, since K-means is just considering a fixed covariance, with this dataset you will likely get this solution.

Improvements

- Use K-means clustering only if there is lot of data available;
- Use median instead of mean;
- Define better distance functions;

K-means still uses only a portion of the parameters of the model

Latent variables

Let's introduce the concept of latent variables, useful to describe the problem but not part of the dataset. The latent variables will express which gaussian has generated which point.

Introduce latent variable $z_k \in \{0,1\}$ (boolean) with $z = (z_1, \dots, z_K)^T$, one for each gaussian, using a 1-out-of-K encoding: z_k is 1 if the point has been generated by gaussian k and all the other elements $z_1, \dots, z_{k-1}, z_{k+1}, \dots, z_K$ are 0.

$P(z)$

$$P(z_k == 1) = \pi_k$$

→ Probability that z_k is 1 corresponds to the prior probability π_k = probability of selecting gaussian k before looking at the data.

Since we assume that the samples are independently distributed, the probability of the vector z is just the product of the probabilities $\pi_k^{z_k}$

$$P(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$

$P(x|z)$

If we know z, we can express the conditional probability of x

$$P(x | z_k = 1) = \mathcal{N}(x; \mu_k, \Sigma_k)$$

→ we know gaussian k is the one that produces x

Thus (compact notation, putting together this notion for all the K gaussians):

$$P(x|z) = \prod_{k=1}^{K} \mathcal{N}(x; \mu_k, \Sigma_k)^{z_k}$$

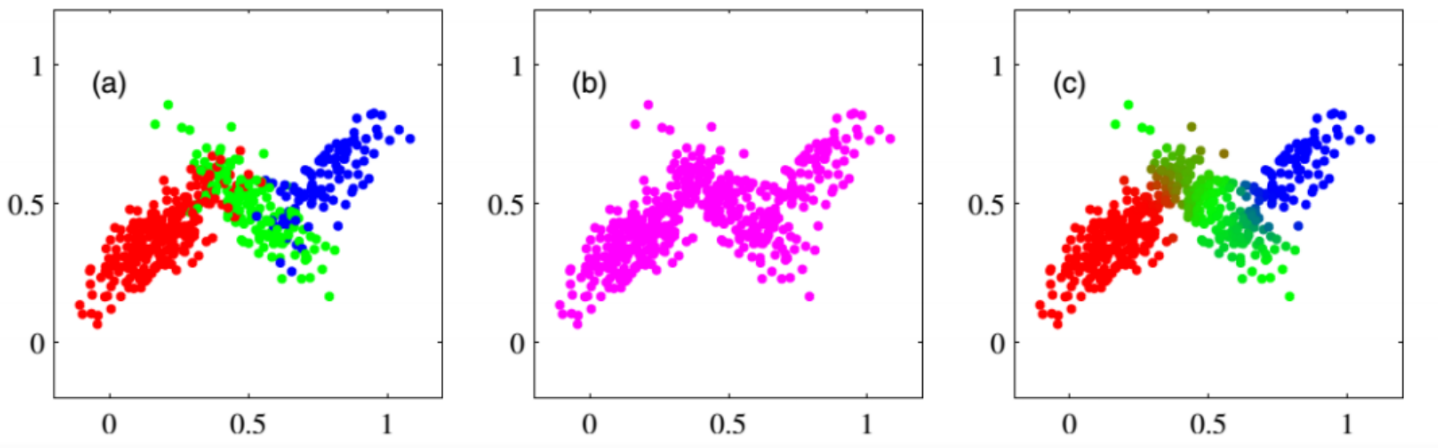
z_k is 1 only for one gaussian

$P(x,z)$

We also know that (joint distribution):

$$P(x,z) = P(x|z)P(z) \longrightarrow \text{chain rule}$$

Joint probability distribution for a set of random variables gives the probability of every atomic joint event on those random variables.



ⓐ $p(x,z)$ with 3 latent variables z (red, green, blue)
ⓑ $p(x)$ marginalized distribution
ⓒ $\gamma(z_{n,k})$ posterior distribution

ⓑ is the input to the model, ⓒ is the result, ⓐ is the ground thruth.

Now what we can do is to express $P(x)$

$$P(x) = \sum_z P(z)P(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

probability of x is, for all the gaussians, the probability that x is generated by that gaussian times the probability of that gaussian

$\sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$ is the Gaussian Mixture Model

Our distribution $P(x)$ can be seen as the marginalization of a distribution $P(x,z)$ over variables z.

$$\sum_z P(z)P(x|z)$$

Marginalization: given a known joint distribution of two discrete random variables X and Y, the marginal distribution of either variable – X for example – is the probability distribution of x when the values of Y are not taken into account. This can be calculated by summing the joint probability distribution over all values of Y.

$$P_X(x_i) = \sum_j P(x_i, y_j) \quad \text{and} \quad P_Y(y_j) = \sum_i P(x_i, y_j)$$

$y \backslash x$	x_1	x_2	x_3	$P(y)$ ↓
y_1	$4/32$	$2/32$	$1/32$	$8/32$
y_2	$3/32$	$6/32$	$3/32$	$15/32$
y_3	$9/32$	0	0	$9/32$
$P(x)$ →	$16/32$	$8/32$	$4/32$	$32/32$

How can we use this result?

Given observations $\mathcal{D} = \{(x_n)_{n=1}^N\}$, each data point x_n is associated to the corresponding variable z_n, which is unknown. If we had z_n this wouldn't be an unsupervised learning problem.

Let's define the posterior probability (probability of selecting a gaussian after looking the data):

$$\gamma(z_k) = P(z_k = 1|x) = \frac{P(z=1)P(x|z_k=1)}{P(x)} \quad \text{Bayes rule}$$

$$= \frac{\pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x; \mu_j, \Sigma_j)}$$